13 SIMPLICITY

The art of maximising the amount of work not done

This chapter looks at ‘maximising the amount of work not done’ – the tenth Agile principle. Most people interpret this principle in one of two ways. They assume this means one of the following:

1. Focusing on ensuring that only the simplest, leanest and fit-for-purpose product is delivered, especially when considering lifecycle-driven documentation, and only producing what adds value.
2. Focusing on maximising the amount of work not done when creating the product, i.e., focusing on simplicity of delivery.

Both of these interpretations are correct, so in the following we will look at both.

13.1 FIT-FOR-PURPOSE PRODUCTS

This principle has links to subjects such as time management and prioritisation and focuses on producing the simplest product that is appropriate for what the customer wants. This requires a clear business case for all stories that are delivered. This principle is therefore about reducing clutter and keeping the backlog focused on whatever needs to be delivered first (see Section 7.1).

Steven Covey (1989) refers to this concept as his Habit 3 or ‘Put the First things First’. We can see how this is also represented in the Pareto Principle (originated by Koch, 1998), i.e. that 80 per cent of the value/functionality/benefit comes from 20 per cent of the effort put in.

The idea of simplicity in information systems has been both supported and challenged. The most quoted source within the Agile community in favour of simplicity is the Standish Group’s series of reports on IT entitled ‘The chaos report’ (Standish, 2002). The overall summary provided by the 2002 chaos report is clear – when the focus strays from producing fit-for-purpose systems, the analytical evidence shows that people simply don’t use a significant majority of the features that are developed.

The following figures were presented at the ‘XP2002’ conference by Jim Johnson, Chairman of Standish Group (Johnson, 2002). He showed a breakdown of features that are actually used in a typical delivered system:

- Features always used – 7%
- Features often used – 13%
- Features sometimes used – 16%
- Features rarely used – 19%
- Features never used – 45%

Therefore, all Agile frameworks have the concept of a clear definition of what

stories are. All Agile frameworks have the concept of the customer collaborating with the team (see Section 11.1) to ensure alignment and focus. And all Agile frameworks have the concept of producing technically fit-for-purpose products.

13.2 FIT-FOR-PURPOSE DELIVERY

Teams need to ensure that any product is delivered in a fit-for-purpose way. This concept is well expressed in the following Lean software development principles (see Section 14.6):

Eliminate waste The three biggest wastes in software development are extra stories, stories constantly changing and the buffers created by crossing organisation boundaries.

Build in quality If defects are routinely found in the verification process, the development process is defective.

Create knowledge Planning is useful. Learning essential.

Defer commitment Abolish the idea that development should start with a complete specification.

Deliver fast Queues cause products to be passed between steps within the process. Typically there will be rigorous sign-off points to 'protect' the team next in the process against the risk of being blamed if something goes wrong with delivery. Communication will largely be via documentation across each of the teams in the delivery value chain, with the inherent communication difficulties that can cause.

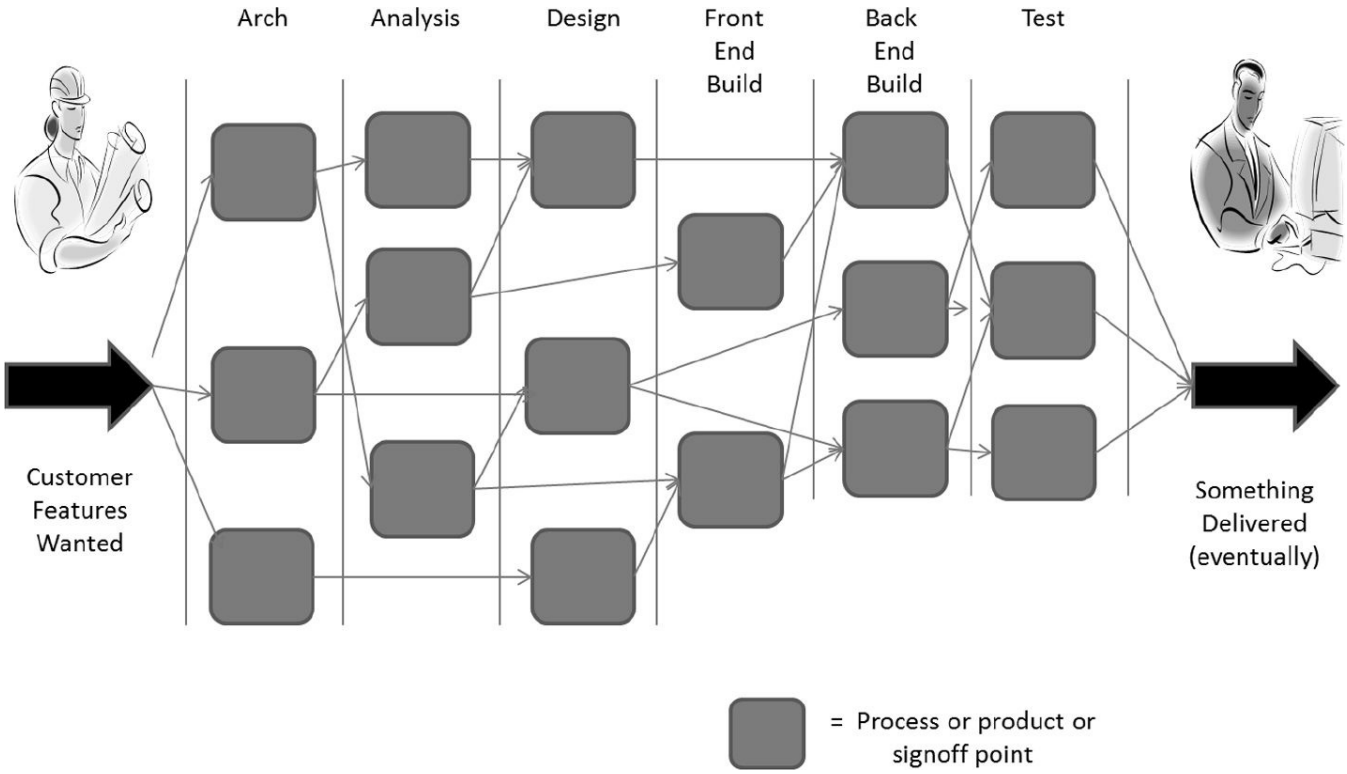
Respect people Engaged, thinking people provide the most sustainable competitive advantage.

Optimise the whole Brilliant products emerge from a unique combination of opportunity and technology when viewed across the whole value chain

Sometimes the way an organisation is structured mirrors the delivery framework being used. Within a Waterfall-shaped organisation the risk is that teams form 'silos' within the key stages of the Waterfall delivery approach – which means that delivering anything across the value chain can become very complicated and full of 'waste' (see Section 6.2).

The risk is that where teams work in completely separated skill areas – e.g. analysts, designers, coders, testers – team members communicate by passing large documents across the 'silos', rather than communicating face-to-face (see Figure 13.1). In a team working with a silo mentality, these unseen boundaries can easily become barriers to effective communication and delivery.

Figure 13.1 Complex silo delivery



This causes the following risks to occur relating to the seven principles:

- **Eliminate waste:**
  - Extra stories – If the customer's experience across the complex silo delivery cycle has been that they don't get everything they want, they are likely to add in 'extra' stories up front with the intent that next time they will get what they want.
  - Stories constantly changing – If the complex silo delivery takes a lot of time it's likely that the customer's requirements will change.
  - Buffers – These are created by crossing organisation boundaries and requiring wait time to get sign-offs associated with crossing the organisational boundary.
- **Build in quality:** The end-to-end quality of the integrated product may not be tested until the end of the lifecycle with the inherent risk that significant functional or technical problems are found at that stage, at which it is too late to do anything about them.
- **Create knowledge:** Learning will typically occur within a siloed delivery organisation; however, it is more difficult to implement end-to-end learning.
- **Defer commitment:** Each silo will typically create detailed documentation that they can pass from one silo to the next. This drives the organisation to make all decisions as early as possible and not align to the concept of last responsible moment (see Section 9.2).
- **Deliver fast:** Queues cause products to be passed between steps within the process, and typically there will be rigorous sign-off points to 'protect' the team next in the process against the risk of being blamed if something goes wrong with delivery. Communication will largely be via documentation across

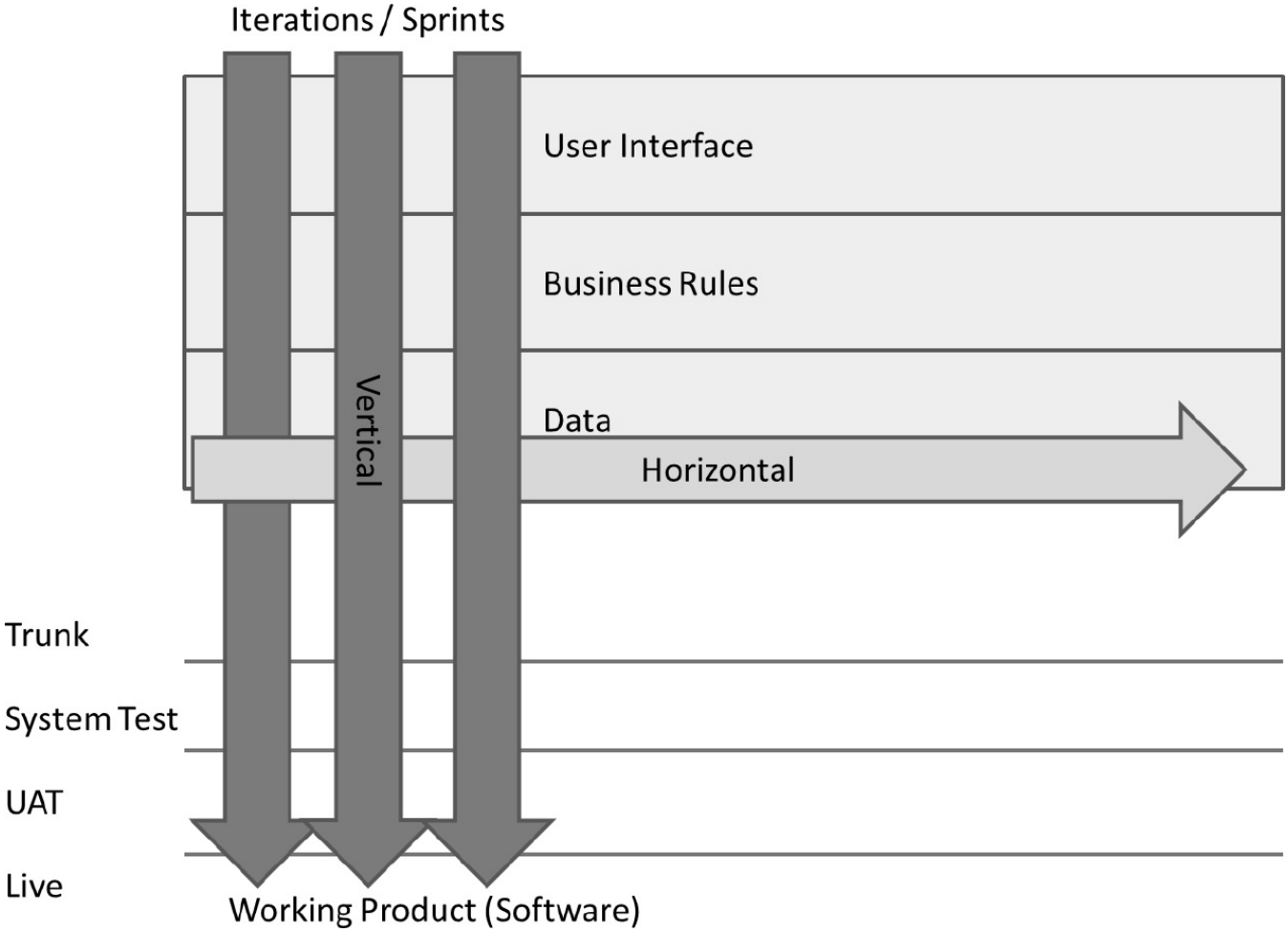
each of the silos within the delivery value chain, with the inherent communication difficulties that causes.

- **Respect people:** The organisation can start to think of the value chain as a product line and the people within the production line as machines; this misses the huge competitive advantage that is provided by engaged people.
- **Optimise the whole:** Silo teams will concentrate on their own silos rather than across the value chain (from concept to cash), which means the required customer focus across the value chain is at risk.

13.2.1 Vertical slices

Vertical slicing is a core concept in all Agile frameworks (see Figure 13.2). Within a traditional delivery approach there will be delivery steps to achieve an outcome (e.g. analysis, design and so on within a Waterfall approach), and this drives horizontal slicing. In an Agile delivery the focus is on producing value-added product for the customer as continuously as possible, which means teams focus on producing the highest value stories in vertical slices down the architecture.

Figure 13.2 Vertical slicing



Agile teams and customers collaborate to produce a single ordered list of what is required for delivery, expressed as stories. This gives a clear and visible indication of the preferred order of delivery and also a clear indication (via the story structure and acceptance criteria) of what makes that feature fit for purpose.

Teams are then structured as feature or component teams (see Section 6.2) so that they have the capability to deliver vertical slices across the value chain. In this case, the impact on the seven principles from Lean software development is as follows:

- **Eliminate waste:**
    - Extra stories – The customer and team continually collaborate to create the appropriate stories, which are delivered within short increments. The customer doesn't need to create extra stories because they are getting continuous delivery.
    - Stories constantly changing – Stories are delivered continuously within short timescales, which means it is unlikely that a story that is in production will change during the short timescale.
    - Buffers – The team works together, not requiring sign-offs between themselves.
- **Build in quality:** All types of testing are fully integrated throughout in all Agile frameworks.
- **Create knowledge:** The team works and continuously learns as a team focused on continuous delivery to the customer.
- **Defer commitment:** Agile aligns to the concept of last responsible moment (see Section 9.2).
- **Deliver fast:** Agile teams are focused on delivery of product increments within short timescales (typically a couple of weeks).
- **Respect people:** Ensuring that this behaviour is part of the organisation culture is one of the responsibilities of the Agile lead (see Section 6.3).
- **Optimise the whole:** Agile teams focus as a team on delivery across the whole value chain delivering value-add product to the customer in short time frames.

Simplicity is about focusing on value and removing all the impediments and disorder that distract from that objective.